

Generalized Perpendicular Bisector and exhaustive discrete circle recognition

Eric Andres, Gaëlle Largeteau-Skapin, Marc Rodríguez

Laboratory XLIM-SIC,
University of Poitiers BP 30179, UMR CNRS 6712
86962 Futuroscope Chasseneuil Cedex, France
{andres,glargeteau,rodriguez}@sic.univ-poitiers.fr

Abstract. This paper presents a generalization of the notion of circum-center as the intersection of perpendicular bisectors. We define Generalized Perpendicular Bisectors between two regions as an area where each point is the center of at least one circle crossing both regions. This allows us to determine all the possible discrete circle centers that cross a given set of pixels. The possible radii can then easily be determined. This exhaustive digital circle parameter computation is adapted to various types of circles/digitization schemes.

1 Introduction

Discrete primitive recognition is one of the longest studied problems in pattern recognition. Recently, within the discrete geometry community, there has been an increased interest for the discrete circle/circular arc recognition problem. This comes partly from the new insight in the analytical description of discrete circles and partly from an increasing interest in differential aspects of discrete analytical geometry. In this paper we are considering the problem of discrete arc and discrete circle recognition from a parameter point of view. We are looking for all the continuous circles that digitized correspond to a given set of pixels. The parameters being the coordinates of the center and the radius. The thickness of the circle can be another parameter that we haven't however addressed in this paper. Classically, parameter space approaches are used for circle recognition [1]. The problem with these Hough transform type approaches is that there are three parameters (abscissa, ordinate of the center and radius) which makes it a three dimensional accumulator matrix. Various methods have been proposed to circumvent this problem [2]. These methods however are adapted for circle recognition in image analysis but not so much for problems such as invertible reconstruction. The discrete geometry community works for many years now on the problem of 2D and 3D invertible reconstruction of discrete objects. Digitization is the transform that associates a discrete object to a continuous one. Reconstruction is the transform that associates a continuous object to a discrete one. A reconstruction method is invertible if the digitization of the reconstructed object is equal to the original discrete object. Straight line segment recognition

has been the main focus of research when it comes to invertible reconstruction [3–6].

There are several aspects to the analytical recognition problem. Firstly, it depends on the digitization method. There is not one type of discrete circles and this of course will change the result. A set of pixels might be a circular arc for one type of discrete circle and not for another. This is often overlooked as most algorithms only focus on one type of circle. Secondly, there are various types of recognition methods that yield different levels of information: the first type of algorithm simply answers the question of whether a given set of pixels corresponds or not to the digitization of a continuous circle. It does not provide any parameters of the continuous circle it may correspond to. The second type of algorithm provides the parameters of one continuous circle corresponding to the set of pixels that is analyzed. A discrete circle can be seen as an equivalence class for all the continuous circle which can be discretized as this discrete circle. This type of algorithm picks one member of this class as answer. Quite often, the answer corresponds to the circle with an integer coordinate center and integer radius. Many approaches consist in searching for a 3D point belonging to the intersection of $2n$ half-spaces in the parameters space, that is the (C_x, C_y, r) -space where (C_x, C_y) is the center of the circle and r the radius [7–9]. Other papers project into the (C_x, C_y, r) -plane. The problem consists then in searching for a 2D point belonging to the intersection of n^2 half-planes (let us cite [10, 11] among the different papers with this approach). When considering the dual of the parameter spaces, the problem corresponds to a separation problem of two sets in 3D by a plane [7, 12]. Many of these approaches may lead to more than one solution even though usually only one solution is given. Providing all the solutions for the recognition problem corresponds to the third type of methods: all the possible centers and all the corresponding radii of the continuous circles that discretized contain the set of analyzed pixels. This is what we propose in this paper and to the best author’s knowledge, this has not been proposed before. The starting point of this work is the definition of a perpendicular bisector for discrete space. In Euclidean geometry, the intersection of the perpendicular bisectors of three points defines the circumcenter of the circumcircle. In this paper we are considering the problem of defining a perpendicular bisector adapted to discrete space for discrete circle recognition purposes. Bisectors appear several times in the literature with usually a definition based on notion of equidistance – from the classical definition to discrete bisector functions which are used to analyze and filter medial axis [13, 14] where the medial axis of a Jordan curve is in any point equidistant to its borders. Bisectors between points and curves or between two curves have also been discussed in detail in the literature [15, 16] but to our knowledge, no definition for the bisector between two coplanar surfaces such as pixels has been proposed so far.

There are various ways of considering points in discrete geometry. This is directly related to the way a continuous object is digitized. We are going to consider here digitization schemes based on distances (in our case the Euclidean, the Manhattan and the Tchebychev distances). The unit spheres for these distances

define regions which we will use to define our Generalized Perpendicular Bisector notion. We are therefore extending the notion of perpendicular bisector as the bisector between two regions. The Generalized Perpendicular Bisector of two 2D regions A and B is defined as the union of the perpendicular bisectors of all the couple of points (p, q) where p and q are respectively points of the regions A and B . This new definition extends the main property of the Euclidean perpendicular bisector of two points that is to be equidistant to both points. However, contrary to a Voronoï diagram, the Generalized Perpendicular Bisector is a surface and not simply a line. The new definition also extends the circumcircle property. For n regions, the intersection of all the Generalized Perpendicular Bisectors defines the Generalized Circumcenter and corresponds to the center of all the circles that cross all n regions. The definition is general and fits all type of regions and can be easily extended to higher dimensions. It gets however complicated very fast as illustrated in the paper with the Generalized Perpendicular Bisector of two disks for the Pythagorean digitization scheme which is a hyperbola. Even for pixels, the Generalized Perpendicular Bisector is a complicated region bordered by straight line segments and pieces of parabolas. We propose a simplification that can be used for a more practical approach in circle recognition.

The starting point of this paper is the definition of the Generalized Perpendicular Bisector and Generalized Circumcenter between two regions. We examine the properties that are extended from usual perpendicular bisectors. In section three, we consider the problem of circle recognition and exhaustive computation of the circle parameters. Different digitization schemes are considered. We then conclude with some open questions and perspectives.

2 The Generalized Perpendicular Bisector and Generalized Circumcenter

In this section we will propose a generalization of the notion of perpendicular bisector and the related circumcenter. Several properties of the Generalized Perpendicular Bisector (*GPB*) and the Generalized Circumcenter (*GC*) are proposed.

2.1 Definitions

The idea is to extend the perpendicular bisector definition to regions. The perpendicular bisector of two points p and q in \mathbb{R}^n corresponds to all the points that are equidistant to both points. In nD , the perpendicular bisector is a hyperplane that is perpendicular to the straight line pq and that passes through its midpoint. The idea here is to replace p and q by regions. This defines the Generalized Perpendicular Bisector (see Fig.1). In our case we are considering finite connected regions. Most of what is presented here is also valid for infinite or disconnected regions. However, as we are going to focus on regions that are around pixels in images, considering that the regions are finite and connected is not very restrictive.

Definition 1 (Generalized perpendicular bisector). Let \mathcal{S}_1 and \mathcal{S}_2 be two connected finite regions. The Generalized Perpendicular Bisector (GPB) of \mathcal{S}_1 and \mathcal{S}_2 is the union of all the perpendicular bisectors of each couple of point (X, Y) where $X \in \mathcal{S}_1$ and $Y \in \mathcal{S}_2$.

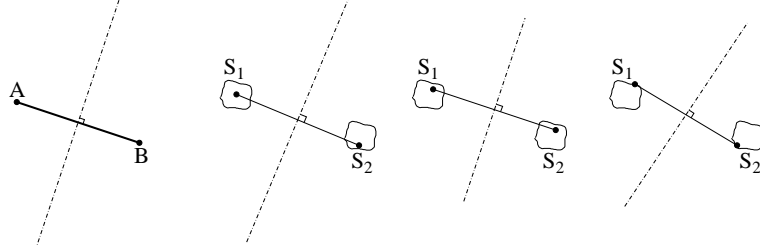


Fig. 1. A Euclidean couple of points has only one perpendicular bisector but a couple of regions has an infinity of bisectors.

For two points p and q , a point r of the perpendicular bisector is equidistant to both points p and q and r is therefore the center of a hypersphere that passes through both points. This leads to an alternative definition of the Generalized Perpendicular Bisector:

Definition 2 (Alternative definition of the Generalized Perpendicular Bisector). Let \mathcal{S}_1 and \mathcal{S}_2 be two finite connected regions. Let $d_i(X) = \min(d(X, \mathcal{S}_i))$ and $D_i(X) = \max(d(X, \mathcal{S}_i))$ where d is the classical Euclidean distance. Every Euclidean point $X \in \mathbb{R}^n$ such that the intervals:

$$[d_1(X), D_1(X)] \cap [d_2(X), D_2(X)] \neq \emptyset$$

belongs to a region called the Generalized Perpendicular Bisector of \mathcal{S}_1 and \mathcal{S}_2 .

These two definitions are equivalent and yield the same area. This second definition shows that, for each point of the Generalized Perpendicular Bisector, there is, in general, an interval of radii for which hyperspheres centered on the point will cut both regions.

The Euclidean circumcenter of a set of Euclidean points is defined by the intersection of all the perpendicular bisectors from pairing the points two by two. Its main property is to be the center of the circle passing through all the points of the set. This naturally extends for Generalized Perpendicular Bisectors.

Definition 3 (Generalized Circumcenter (GC) of a finite set of regions). The Generalized Circumcenter (GC) of set of n finite and connected regions $\mathcal{S} = (\mathcal{S}_i)_{i \in [1, n]}$ is the intersection of the Generalized Perpendicular Bisectors (GPB) of every two regions of the set:

$$GC(\mathcal{S}) = \bigcap_{i, j \in [1, n], i < j} (GPB(\mathcal{S}_i, \mathcal{S}_j)).$$

The *GC* is defined as the intersection of the *GPB* of a finite set of regions. One can consider infinite sets of regions without much of a problem (one has to be a little bit careful when using the Helly Theorem in proof of Theorem 1). In practice however the sets are always finite. We can see on figure 2 the Generalized Circumcenter of three pixels (square surfaces) and on figure 3 the Generalized Circumcenter of four pixels. In this case we are considering the Supercover/Standard digitization model [17]. Figure 4 show all the circles that cross the pixels $(0,0)$, $(2,6)$ and $(6,3)$.

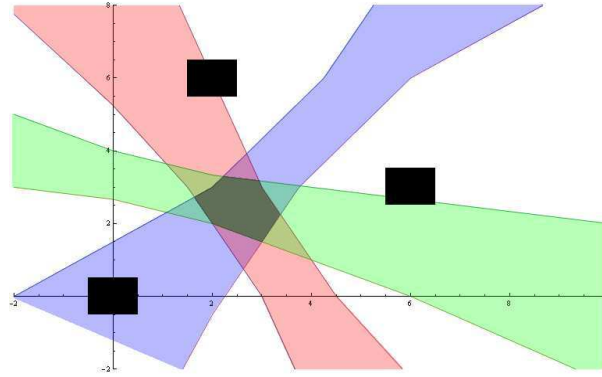


Fig. 2. The Generalized Perpendicular Bisector and Generalized Circumcenter of $\{(0,0), (2,6), (6,3)\}$.

2.2 Properties

The Generalized Perpendicular Bisector (*GPB*) of two finite connected regions is not a straight line but an infinite set of straight lines that forms a region which is connected, infinite and usually not convex. The shape of the Generalized Perpendicular Bisector depends on the shape of both regions. For example, let $C_1 = (O_1, r_1)$ and $C_2 = (O_2, r_2)$ be two disks. The Generalized Perpendicular Bisector of both discs is a region bordered by a hyperbola (see Fig.10). The *GPB* is in general a complicated object as we will see for two square regions in the Supercover or naive digitization. The *GPB* has already been studied usually with regions such as point/curve, curve/curve [15]. A particular application field can be found in geometric design [18].

From the definition of the *GPB* we can deduce some immediate properties:

Property 1: The Generalized Perpendicular Bisector of two overlapping regions is the whole space even if they only share one point.

Proof. The bisector of two identical points is not defined if we consider the perpendicular line passing through the midpoint of the two points. If we consider

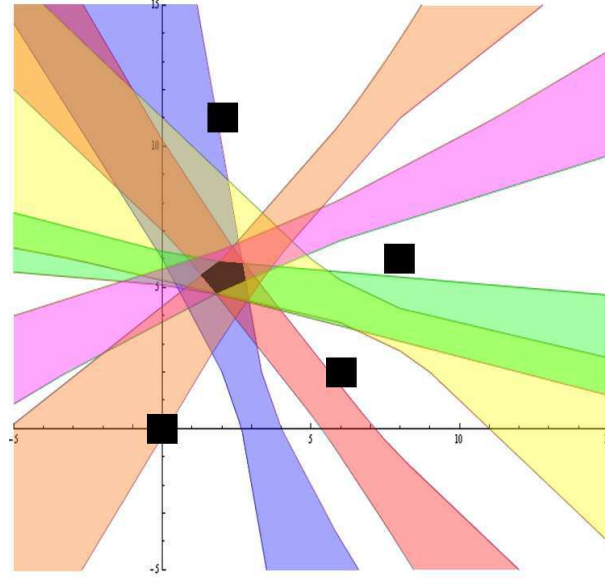


Fig. 3. The Generalized Perpendicular Bisector and Generalized Circumcenter of $\{(0, 0), (2, 11), (6, 2), (8, 6)\}$.

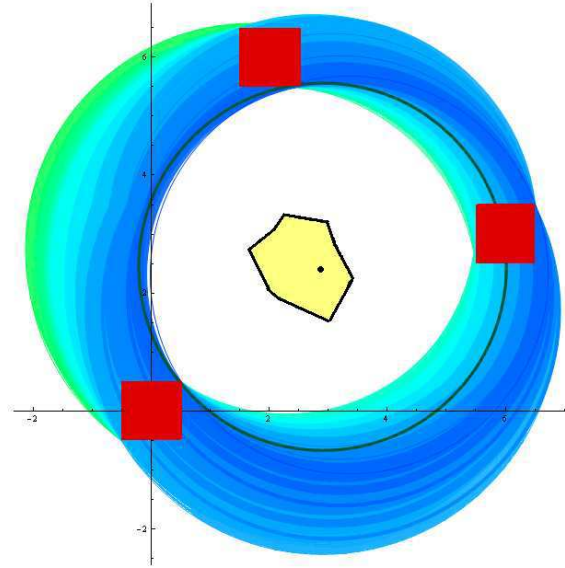


Fig. 4. The pixels $\{(0, 0), (2, 6), (6, 3)\}$, the GC and all the circles that cross those pixels. In black, one example of such a circle (its center is the black dot in the GC).

the equidistance definition however it is easy to see that every point of the space is equidistant to both points. \square

Property 2: Each point of the Generalized Perpendicular Bisector of two regions is the center of a hypersphere crossing both regions.

These two properties are a direct consequence of the Generalized Perpendicular Bisector definition and have already been proposed in [19].

An important property of the Generalized Circumcenter is the existence of a circle that intersects all the regions at once. All the points in the Generalized Circumcenter of a set of a finite, connected regions are the center of at least one circle crossing every two regions of the set. That does not automatically mean that there exists a circle (i.e. in this case we are looking for a radius) that crosses every region of the set. Actually it does as the following theorem shows:

Theorem 1 (Hypersphere existence). *If the Generalized Circumcenter $GC(\mathcal{S})$ of a finite set of regions \mathcal{S} is not empty, then for all $X \in GC(\mathcal{S})$ there exists at least one hypersphere centered in X that crosses all the regions of the set \mathcal{S} .*

Proof. This result is the direct application of Helly's theorem [20]. Let us briefly recall what Helly's theorem says: Let us suppose that we have a finite collection of n convex subsets of \mathbb{R}^d with $n > d$. If the intersection of every $d + 1$ of these sets is nonempty, then the whole collection has a nonempty intersection. In our case we are looking for a common radius R such that there is a circle centered in a point $X \in GC(\mathcal{S})$ with radius R that crosses all the regions of \mathcal{S} . Let us consider a point $X \in GC(\mathcal{S})$. For each region S_i of \mathcal{S} , there is a closed interval of radii $R_i = [dmin(X, S_i), dmax(X, S_i)]$ with $dmin(X, S_i)$ the minimal distance between X and S_i and $dmax(X, S_i)$ the maximal distance between X and S_i . X is point of $GC(\mathcal{S})$ and therefore every two intervals R_i and R_j with $i < j$ have an intersection. We can therefore apply Helly's theorem in \mathbb{R} which proves our theorem. \square

Property 3: The Generalized Circumcenter of a set of regions $\{S_i\}$ is infinite if and only if there exists a hyperplane that cuts all the regions S_i .

Proof. This property is new and quite straightforward as well. If there exists a hyperplane that cuts all the regions then a center that is at the infinite, perpendicular to the hyperplane, defines a hypersphere that cuts all the regions, thus belonging to the Generalized Circumcenter. If such a hyperplane does not exist then a hypersphere that cuts all regions has necessarily a finite radius and thus a center that is not at the infinite making the Generalized Circumcenter finite. \square

An example of an infinite Generalized Circumcenter can be seen in figure 5 with three regions (pixels in this case) and the corresponding Generalized Perpendicular Bisectors.

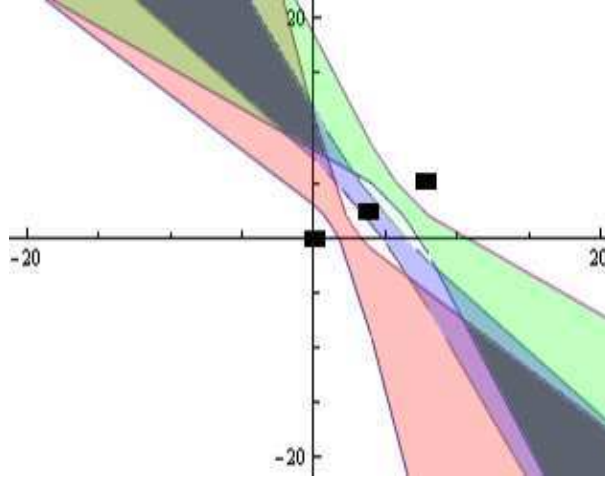


Fig. 5. when the pixels are aligned, the Generalized Circumcenter (dark region) is infinite and not connected. The pixels are marked in black.

3 Circle Recognition and Parameter Estimation

We are now considering objects from discrete spaces. While the Generalized Perpendicular Bisector is defined in arbitrary dimensions, we are going to focus on the dimension two. We are going to consider regions that correspond to various discrete analytical models: squares for Naïve (such as the Bresenham circle [21]) and Supercover circles, circular disks for Andres circles [22]. We are considering various types of circles because it is important that, when recognizing a discrete primitive and estimating its parameters, we know what we are actually recognizing. There are various types of discrete circles from different analytical models. Our recognition and parameter estimation method works for all known classical type of discrete circles. Let us first recall some elements on the different type of circles we are considering here.

3.1 Recall on discrete analytical models

We are going to consider here discrete analytical models based on a distance d . Let us consider a Euclidean object E . The digitization $D_d(E)$ of E according to the discrete analytical model associated to d is defined by:

$$D_d(E) = \left\{ p \in \mathbb{Z}^2 \mid d(p, E) \leq \frac{1}{2} \right\}.$$

This type of digitizations are particularly interesting because they form coherent digitization models with a global definition. For classical distances such as

the Manhattan distance d_1 , the Euclidean distance d_2 or the Tchebychev distance d_∞ , it allows to describe the primitives analytically with a global definition. There are also some basic definitions that are very useful when constructing objects in the discrete world such as, for E, F two Euclidean objects, $D_d(E \cup F) = D_d(E) \cup D_d(F)$ (see [17] for more details on the Supercover analytical model).

There is an equivalent definition for the digitization scheme where regions appear. An equivalent definition is based on the unit sphere $B_d(1)$ of diameter one for the distance d . The discretization can be written as follow:

$$D_d(E) = \{p \in \mathbb{Z}^2 \mid (B_d(1) \oplus p) \cap E \neq \emptyset\} \text{ where } B_d(1) \text{ is the pixel region.}$$

and $A \oplus B = a + b, a \in A, b \in B$ is the Minkowski sum.

The pixel region is the region that is associated to each pixel of the discrete circle we are trying to analyze. The discretization models we are going to consider are, firstly the Supercover/Standard models based on the distance d_∞ . Then the Naïve model which is, in 2D, from the computation point of view similar to the Supercover/Standard models and then we will mainly focus on the Pythagorean model based on the distance d_2 . Let us start with the general principal of the circle recognition and parameter estimation algorithm.

3.2 Circle/circular arc recognition and parameter estimation

From the properties of the Generalized Perpendicular Bisector and the Generalized Circumcenter, for a given circle type, we can give the general idea for a circle/circular arc recognition algorithm that provides all the parameters of the discrete circle/circular arc. By parameter estimation, we mean all the possible centers, the Generalized Circumcenter region, from which one can compute all the possible radii of the Euclidean circles that, once digitized, contain the set of pixels we were analyzing. We do not differentiate here the recognition of a circle or a circular arc. There is of course a difference but not a very important one from the algorithmic point of view. Once the circle parameters are computed for a circular arc, the end points of the arc are easy to compute. As we mentioned in the previous section, the circle/circular arc recognition and parameter estimation depends on the type of circles we are considering. A same set of pixels can be recognized as a circle/circular arc of a certain type and not a circle/circular arc of another type or be recognized as circles/circular arcs for different type of circles/circular arcs but with different parameters.

We are now considering that we try to recognize a specific type of circles. For a discrete point p , $Region(p)$ corresponds to the pixel region associated to the digitization model for the type of circle we are considering (see algorithm 1).

The complexity of the algorithm depends on the complexity of the pixel region intersections times n^2 . This doesn't make it a very efficient algorithm. It yields however the Generalized Circumcenter region where all the centers of all the circles that cut all the regions corresponding to the pixels of the discrete circle. If the Generalized Circumcenter is not empty then we have recognized a

Algorithm 1: Circle Recognition and Center localization algorithm

```

input : A set  $P = p_i$  of  $n$  pixels
output: The Generalized Circumcenter
begin
  initialize  $GenCircumcenter = \mathbb{R}^2$ ;
  for  $i = 1, \dots, n - 1$  do
    for  $j = i + 1, \dots, n$  do
      if  $Region(p_i) \cap Region(p_j) \neq \emptyset$  then
         $GenCircumcenter = GenCircumcenter \cap GPB(p_i, p_j)$ ;
  return  $GenCircumcenter$ 

```

Algorithm 2: Radii Estimation for a point r with a set of pixels P

```

input : A point  $c \in \mathbb{R}^2$ , a set of pixels  $P = \{p_i\}_{i \in [1, n]}$ 
output: The radii interval  $[r_{min}, r_{max}]$  for a circle centered in  $c$  passing through
  all the pixels of  $P$ 
begin
  initialize  $Radius\_Interval = \mathbb{R}$ ;
  for  $i = 1, \dots, n$  do
     $r_{min} = smallest\_distance(c, p_i)$ ;
     $r_{max} = biggest\_distance(c, p_i)$ ;
     $Radius\_Interval = Radius\_Interval \cap [r_{min}, r_{max}]$ ;
  return  $Radius\_Interval$ 

```

discrete circle of the considered type otherwise the set of pixels does not correspond to such a circle. The previous algorithm provides a region where all the centers are located but does not provide the corresponding radii. The radii can be computed easily for a given point in the Generalized Circumcenter (see algorithm 2).

Except of course in particular complicated cases, the computation of the smallest distance and the biggest distance between a point and a pixel region should be done in constant time. The algorithm to determine the radii for a given point c can thus be conducted in linear time. Note that if we want to know if a specific point c can be the center of a circle for the set of pixels we are analyzing, the circle recognition and the radii estimation can be conducted by the “Radii Estimation” algorithm. If the radii interval is empty then the set of pixels does not correspond to any circle of center c .

Let us now examine different circle types and see how this influences the complexity of these algorithms. Let us first consider circle definitions based on the Tchebychev distance d_∞ .

3.3 Supercover and Standard circle recognition and parameter estimation

In this section we are considering the Supercover or Standard digitization models. These digitization models have been studied in [17, 23]. The Standard digitization model is simply a variant without bubbles of the Supercover model. They are based on the Tchebychev distance d_∞ and the corresponding pixel region is a square of side one, exactly a pixel. This makes the Supercover discretization model known for a some time [24]. The discretization of a Euclidean object corresponds to the center of all the pixels this object cuts. In the following, we explain how the Generalized Perpendicular Bisector between two pixels can be piecewise computed (here pixels and pixel region are the same).

Let $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ be two pixels of which we want to compute the Generalized Perpendicular Bisector area. The pixel region is the unit square region centered on a \mathbb{Z}^2 point: $\{(x, y), x \in [x_i - 0.5, x_i + 0.5] \text{ and } y \in [y_i - 0.5, y_i + 0.5]\}$. Let $X(x, y)$ be another point in \mathbb{R}^2 . To know if the point X is a good candidate to be in the Generalized Perpendicular Bisector area, we have to compute $d_1(X) = \min(d(X, P_1))$, $D_1(X) = \max(d(X, P_1))$, $d_2(X) = \min(d(X, P_2))$ and $D_2(X) = \max(d(X, P_2))$.

We can see on figure 6 that the point of the pixel P that is closest to X can be either: X itself if X belongs to the pixel P , a point on an edge of P if X can be orthogonally projected on an edge of P or a vertex of P in all the other cases. The point F of the pixel P that is farthest from X is always a vertex of P . Each pixel splits the space into 16 areas and so two non aligned pixels divide space in at most 49 areas where each area has a different formula for $d_1(X)$, $d_2(X)$, $D_1(X)$, and $D_2(X)$ (with of course symmetries).

Let us now examine what the intersection of intervals means in terms of equations: $[d_1(X), D_1(X)] \cap [d_2(X), D_2(X)] \neq \emptyset$. It is easier to verify the conditions where a point does not belong to a Generalized Perpendicular Bisector which occurs when $D_1(X) < d_2(X)$ or $D_2(X) < d_1(X)$.

Let $C_i(C_{ix}, C_{iy})$ be the closest point of X in the pixel P_i and let $F_i(F_{ix}, F_{iy})$ be the furthest point of X in P_i . In each area, a constraint $d_i \leq D_j$ can be written:

$$\sqrt{(x - C_{ix})^2 + (y - C_{iy})^2} \leq \sqrt{(x - F_{jx})^2 + (y - F_{jy})^2}$$

where $C_{ix} \in \{x, x_i + 0.5, x_i - 0.5\}$, $C_{iy} \in \{y, y_i + 0.5, y_i - 0.5\}$, $F_{jx} \in \{x, x_j + 0.5, x_j - 0.5\}$ and $F_{jy} \in \{y, y_j + 0.5, y_j - 0.5\}$.

For the areas where X is not orthogonally projected on an edge of a pixel, we know that $x \neq C_{ix}$, $x \neq F_{jx}$, $y \neq C_{iy}$, $y \neq F_{jy}$. The two constraints can then be reduced to half planes inequations of type $\alpha x + \beta y + \gamma \leq 0$ where $\alpha, \beta, \gamma \in \mathbb{R}$.

For example, for the first case of Fig.6, we have $C_1 = (x_1 + 0.5, y_1 - 0.5)$, $F_1 = (x_1 - 0.5, y_1 + 0.5)$ and $C_2 = (x_2 - 0.5, y_2 + 0.5)$, $F_2 = (x_2 + 0.5, y_2 - 0.5)$.

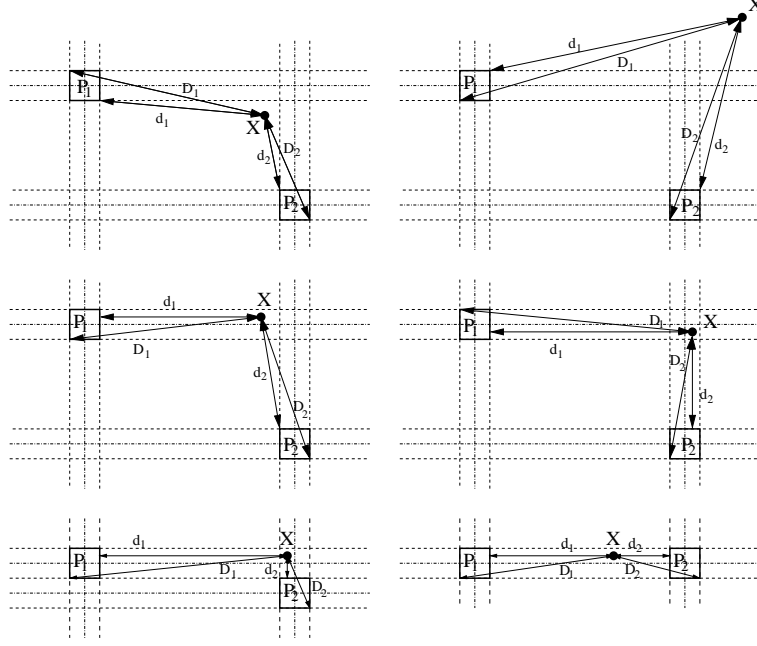


Fig. 6. X is a possible circle center if $[d_1(X), D_1(X)] \cap [d_2(X), D_2(X)] \neq \emptyset$.

The distance constraint $d_1 \leq D_2$ is therefore :

$$\sqrt{(x - (x_1 + 0, 5))^2 + (y - (y_1 - 0, 5))^2} \leq \sqrt{(x - (x_2 + 0, 5))^2 + (y - (y_2 - 0, 5))^2}$$

$$\Leftrightarrow 0 \geq 2(x_2 - x_1)x + 2(y_2 - y_1)y + (x_1^2 - x_2^2 + x_1 - x_2 + y_1^2 - y_2^2 - y_1 + y_2).$$

In the other areas, we are at least in one of these cases $C_{i_x} = x$, $C_{j_x} = x$, $C_{i_y} = y$ or $C_{j_y} = y$. There are no simplifications during the calculation, the constraints correspond to pieces of parabolas of the form:

$$(\alpha_1 x + \alpha_2 y)^2 + \beta x + \gamma y + \delta \leq 0 \text{ where } \alpha_1 = 0 \text{ or } \alpha_2 = 0.$$

Figure 2 shows an example of circumcenter of three regions defined by the Tchebychev distance (unitary squared regions). Figure 4 shows all the corresponding circles. Figure 7 illustrates the GC with the maximal radius at each point for a set of pixels that corresponds to a Bresenham circle. Note that a Bresenham circle is only a subset of Supercover circles. The Supercover digitization of all circles corresponding to the GC contains those pixels and some more. We used a Bresenham circle in order to be able to compare the results obtained with the d_∞ , d_2 and d_1 distance as the pixel region of the d_1 distance is the smallest. Compare with figures 9 and figure 11.

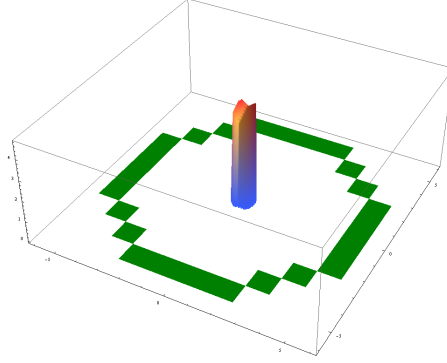


Fig. 7. A Bresenham circle of radius 5. Using the Tchebychev distance, in the center of the image, the Generalized Circumcenter with the maximum radius for each point (shown in the z direction)

3.4 Naïve circle recognition and parameter estimation

In this section we are considering the Naïve digitization model. This model is based on the Manhattan distance d_1 . In 2D, the unit sphere for this distance is a square of side $\frac{\sqrt{2}}{2}$ rotated by 45 degrees. It corresponds to 8-connected objects. The Bresenham circle [21] corresponds to a Naïve digitization of a continuous circle (see figure 12).

The Generalized Perpendicular Bisector of two pixels considering the Naïve model can be computed using the formulas defined in the previous part (for the Supercover and the Standard model). A rotation (of angle $\frac{\pi}{4}$) and a rescaling (of factor $\frac{\sqrt{2}}{2}$) lead to the expected result.

Figure 8 shows the recognition of a Naïve circle. We can see that the Generalized Circumcenter contains all the centers of all the circles crossing all the regions even if the circle is not centered on a pixel.

Figure 9 shows the shape of the surface corresponding to the minimal and maximal radius for a given point of the Generalized Circumcenter. As we can see, those surfaces are not simple. This leaves one open question: is there a radius valid for all the Generalized Circumcenter points?

In higher dimensions, this result no longer stands: for example, the unit 3D sphere for the Manhattan distance is an octagon which is obviously not the rotation of a cube.

3.5 Andres circle recognition and parameter estimation

There is a class of circles that have a definition based on the Euclidean distance. The definition has been proposed in all dimensions by E. Andres [22], one of the authors of this paper. The 2D Andres circles are defined as follows:

$(x, y) \in \mathbb{Z}^2$ belongs to the Andres circle of center (x_o, y_o) and radius R and

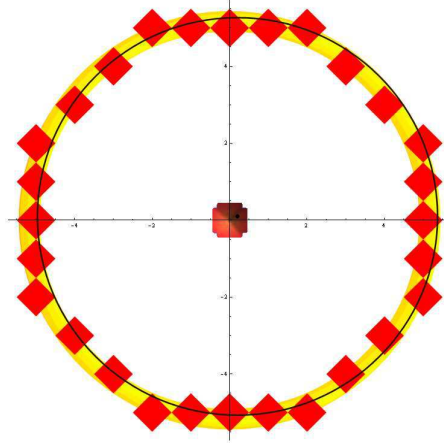


Fig. 8. This figure shows the Bresenham circle of radius 5 and center $(0,0)$ (in dark grey, the pixel regions). In light grey all the possible circles that once digitized are equal to the Bresenham circle. In black one such continuous circle with its center (black dot in the middle). In the middle of the figure, the Generalized Circumcenter.

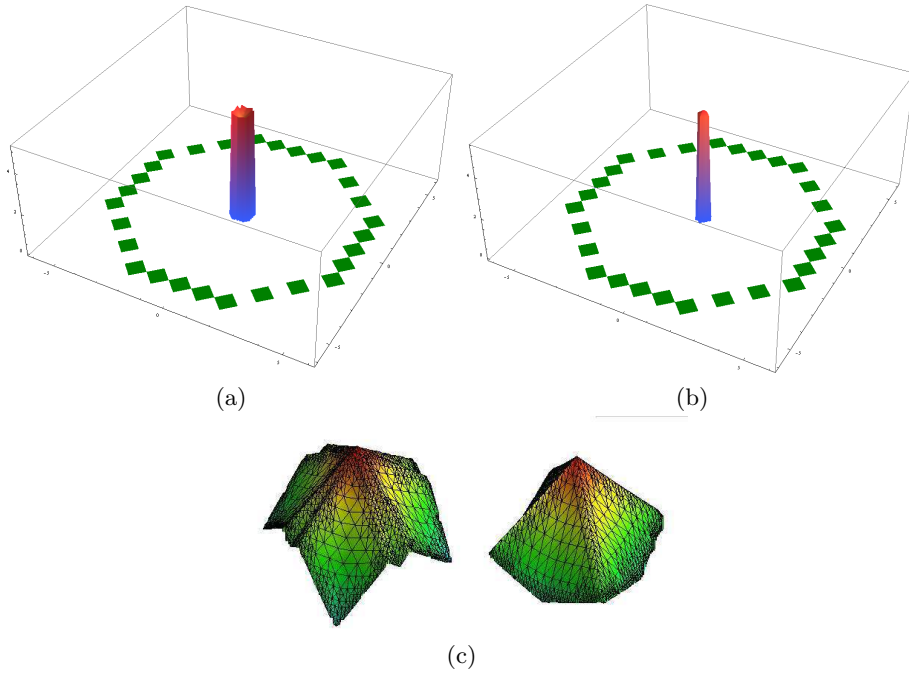


Fig. 9. Generalized Circumcenter for a Naïve circle recognition. (a) minimum radius for each point of the circumcenter. (b) maximum radius for each point of the circumcenter. (c) shapes of surfaces corresponding to the maximum and minimum radii.

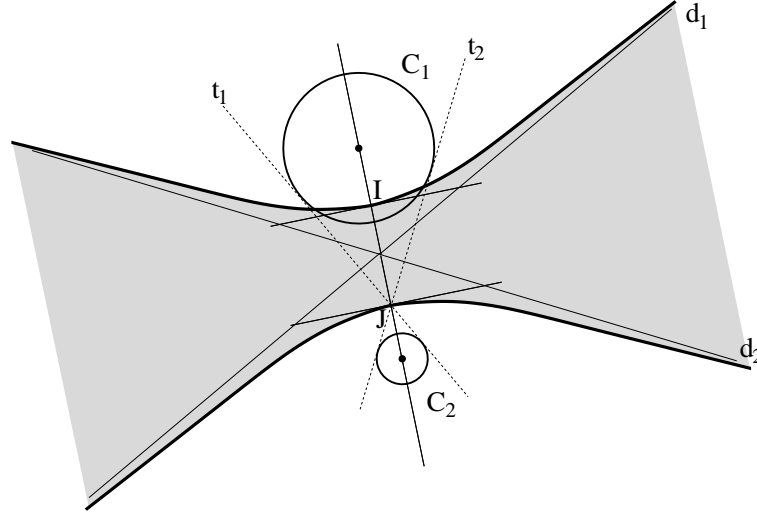


Fig. 10. Example of a Generalized Perpendicular Bisector of two discs.

thickness ω if and only if:

$$\left(R - \frac{\omega}{2}\right)^2 \leq (x - x_o)^2 + (y - y_o)^2 < \left(R + \frac{\omega}{2}\right)^2$$

Note that in this definition, x_o, y_o, R and ω have no particular reason to be integers. It is easy to see that this type of circles is associated to the distance d_2 . Another way of defining such a circle is indeed:

$$\left\{ (x, y) \in \mathbb{Z}^2 \mid C(x_o, y_o, R) \cap B_2\left(\frac{\omega}{2}\right) \neq \emptyset \right\}$$

where $C(x_o, y_o, R)$ is the Euclidean circle of center (x_o, y_o) and radius R .

This means that the pixel regions in our case correspond to disks. This is actually a particularly complicated case because the Generalized Perpendicular Bisector of two discs is a hyperbola (see figure 10). The computation of the Generalized Circumcenter requires therefore that we intersect hyperbolas.

Figure 11 illustrates a recognition for a set of pixels that corresponds to a Bresenham circle of radius 5. The disks correspond to all the pixel regions for this circle. In the center of the image we have represented the Generalized Circumcenter with the maximum radius for each point (shown in the z direction).

4 Conclusion and Perspectives

In this paper, we present the generalization of perpendicular bisectors and circumcenters to regions in an arbitrary dimensional space. This extension is very interesting because many properties from classical perpendicular bisectors are

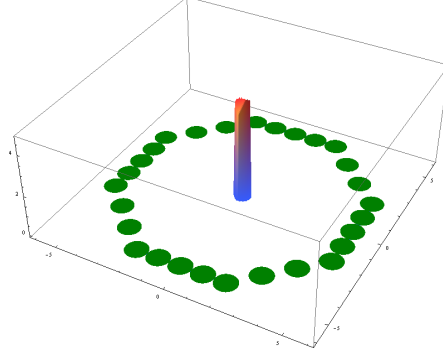


Fig. 11. Circumcenter of a Bresenham circle of radius 5 with the Euclidean distance.

preserved. We have shown that, in 2D, all the points of the Generalized Circumcenter of a set of regions are the center of at least one circle that crosses all the regions of the set.

The Generalized Circumcenter is a new interesting approach to digital circles recognition and reconstruction because it contains all the centers of all the Euclidean circles that pass through all the regions. It can be used for full discrete circle recognition or only for discrete circular arc recognition.

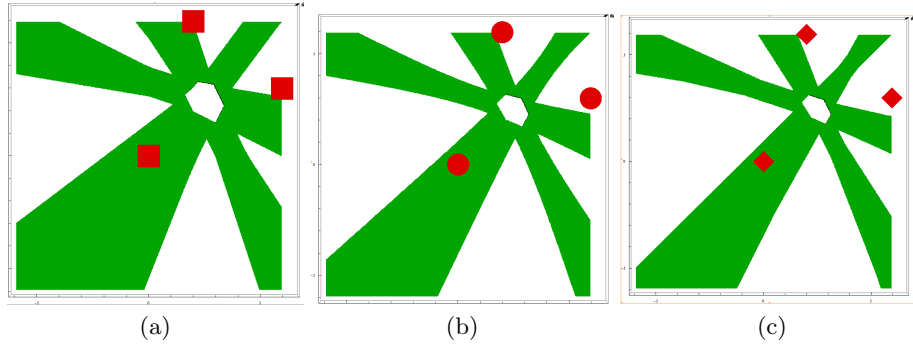


Fig. 12. Generalized Perpendicular Bisectors and Generalized Circumcenter of three unit spheres considering the Tchebychev distance (a), the Euclidean distance (b) and the Manhattan distance (c).

As an application, we have described the Generalized Perpendicular Bisector of two square regions. This naturally leads to the recognition of Supercover and Standard digital circles since the square is the unit sphere for the distance d_∞ . Given a set of pixels S , we are therefore able to provide the exhaustive set of possible Euclidean circles (centers and radii) whose Supercover digitizations is S .

We also present similar results for the Naïve digitization model which uses the Manhattan distance d_1 (regions are rhombus of side $\frac{\sqrt{2}}{2}$) and for the Pythagorean digitization model which uses the Euclidean distance d_2 (regions are disks).

In this last case, however, the algorithm can not be easily implemented since we need to intersect hyperbolas. In the two first cases, we have shown that the Generalized Perpendicular Bisector can be approximates using only straight segments.

Much work remains to be done however. There are some complicated algorithmic problems that need to be solved before an efficient circle recognition algorithm can be proposed. The intersection of the simplified Generalized Perpendicular Bisectors is algorithmically costly because (despite the fact that they are defined by straight lines) they are not convex. Secondly, for n regions we have $O(n^2)$ Generalized Perpendicular Bisectors to intersect in order to compute the Generalized Circumcenter. We are looking right now into some simplification in the recognition process that should allow us, at least, to be able to fast reject cases where there are no solutions. The long term goal of all this work is to be able to have an invertible circle recognition algorithm that can be paired with straight line recognition algorithms. From the theoretical point of view, we would like to see how the Generalized Perpendicular Bisector and the Generalized Circumcenter relate to Voronoï diagrams and medial axis.

One of the last advantages of this approach is that regions do not need to be of the same shape or size. We can therefore imagine to compute the Generalized Perpendicular Bisector between two squares of different sizes. This could allow multiscale or noisy circle/hypersphere recognition [19].

References

1. Ioannou, D., Huda, W., Laine, A.: Circle recognition through a 2d hough transform and radius histogramming. *Image Vision and Computing* **17** (1999) 15–26
2. Gonzalez, R., Woods, R., Eddins, S.: *Digital Image Processing Using MATLAB(R)*. Prentice Hall (2004)
3. Vittone, J., Chassery, J.M.: (n,m)-cubes and farey nets for naive planes understanding. In: *Discrete Geometry for Computer Imagery*. Volume LNCS vol. 1568. (1999) 76–90
4. Vittone, J., Chassery, J.M.: Recognition of digital naive planes and polyhedrization. In: *Discrete Geometry for Computer Imagery*. Volume LNCS vol. 1953. (2000) 296–307
5. Breton, R., Sivignon, I., Dexet, M., Andres, E.: Towards an invertible euclidean reconstruction of a discrete object. In: *Discrete Geometry and Computer Imagery*. Volume LNCS vol. 2886. (2003) 246–256
6. Dexet, M., Andres, E.: A generalized preimage for the digital analytical hyperplane recognition. *Discrete Applied Mathematics* **157** (2009) 476–489
7. O'Rourke, J., Rao Kosaraju, S., Megiddo, N.: Computing circular separability. *Discrete and Computational Geometry* **1** (1986) 105–113 10.1007/BF02187688.
8. Sauer, P.: On the recognition of digital circles in linear time. *Computational Geometry* **2** (1993) 287 – 302

9. Damaschke, P.: The linear time recognition of digital arcs. *Pattern Recogn. Lett.* **16** (1995) 543–548
10. Kovalevsky, V.: New definition and fast recognition of digital straight segments and arcs. (1990) II: 31–34
11. Coeurjolly, D., Gerard, Y., Reveillès, J.P., Tougne, L.: An elementary algorithm for digital arc segmentation. *Discrete Applied Mathematics* **139** (2004) 31–50
12. Roussillon, T., Sivignon, I., Tougne, L.: Test and measure of circularity for digital curves. In: *IPCV*. (2008) 518–524
13. Couprie, M., Coeurjolly, D., Zrour, R.: Discrete bisector function and euclidean skeleton in 2d and 3d. *Image Vision and Computing* **25** (2007) 1543–1556
14. Talbot, H., Vincent, L.: Euclidean skeletons and conditional bisectors. *SPIE* **1818** (1992) 862–876
15. Farouki, R.T., Johnstone, J.K.: Computing point/curve and curve/curve bisectors. In Fisher, R.B., ed.: *The Mathematics of Surfaces V*. Oxford University, Oxford (1994) 327–354
16. Peternell, M.: Geometric properties of bisector surfaces. *GM* **62** (2000) 202–236
17. Andres, E.: Discrete linear objects in dimension n : the standard model. *Graphical Models* **65** (2003) 92–111
18. Farouki, R.T., Ramamurthy, R.: Degenerate point/curve and curve/curve bisectors arising in medial axis computations for planar domains with curved boundaries. *Comput. Aided Geom. Des.* **15** (1998) 615–635
19. Rodríguez, M., Sere, A., Largeteau-Skapin, G., Andres, E.: Generalized perpendicular bisector and circumcenter. In: *CompImage'2010*. (2010) 1–10
20. Helly, E.: Über mengen konvexer körper mit gemeinschaftlichen punkten. In: *Jahresbericht der Deutschen MathematikerVereinigung*. Volume 32. (1923) 175–176
21. Bresenham, J.: A linear algorithm for incremental digital display of circular arcs. *Commun. ACM* **20** (1977) 100–106
22. Andres, E.: Discrete circles, rings and spheres. *Computer & Graphics* **18** (1994) 695–706
23. Andres, E.: The supercover of an m -flat is a discrete analytical object. *Theoretical Computer Science* **406** (2008) 8 – 14 *Discrete Tomography and Digital Geometry: In memory of Attila Kuba*.
24. Cohen Or, D., Kaufman, A.: Fundamentals of surface voxelization. *Graphical Models and Image Processing* **57** (1995) 453–461